ALGOL BULLETIN No. 31

March 1970

CONTENTS PAGE

AB31.0	Editor's Notes		1
AB31.0.1	New Arrangements for AB		1
AB31.0.2	Contributions to AB		2
AB31.0.3	Corrections to AB30.3.2		3
AB31.1	News Items		7
AB31.1.1	Working Group 2.1		7
AB31.1.1.1	"Minority Report"		7
AB31.1.1.2	Report on ALGOL 68		8
AB31.1.1.3	Recent Activities		8
AB31.1.2	ACM Symposium of ALGOL 68		8
AB31.2	Correspondence		9
AB31.2.1	P.J. Barker:	Flow Charting of *for* Statements	9
AB31.2.2	B. Higman:	Simplified Notation	11
AB31.2.3	M. Woodger:	Labels in Array Bounds in ALGOL 60	12
AB31.2.4	B. Randell:	Resignation from WG 2.1	14
AB31.2.5	P.R. Jones:	Parentheses in If-Then Clauses	15
AB31.2.6	S.H. Chisman:	ALGOL Bulletin	15
AB31.3	Working Papers		16
AB31.3.1	P.E. Lauer:	Formal Definition of ALGOL 60 (notice only)	16
AB31.3.2	C. Pair:	Concerning the Syntax of ALGOL 68	16
AB31.3.3	H.J. Bowlden	ALGOL 68 - Comments and Recommendations	28
AB31.3.4	J. Palme	"Atoms" as a Measurement of Program Length	33
AB31.3.5	J. Nievergelt:	Dummy Statements and Consecutive Semicolons in ALGOL 60	35
AB31.3.6	C.H. Lindsey:	An ISO-Code Representation for ALGOL 68	37
AB31.4	References		61
AB31.5	The New Complete Circulation List		63

AB31.3 WORKING PAPERS

AB31.3.1 Formal Definition of ALGOL 60

P.E. Lauer

ABSTRACT

This report constitutes a formal definition of the syntax and semantics of the algorithmic language ALGOL 60. The method is based on an abstract syntax specifying programs in an abstract form. These are interpreted by means of an abstract machine characterized by the set of the states it can assume and its state transition function. An abstract program specifies an initial state of the machine and the subsequent behaviour of the machine is said to define the interpretation of the given abstract program. The concrete representation of abstract programs as character strings is achieved by means of a representation system.

(It was considered that the text of this important document was too long - 62 pages - to be included complete in the AB. Since the circulation list for the AB is currently under revision, it would be appreciated if readers who wish to have a copy of Mr. Lauer's Report would ask him direct. The address is:

Mr. P.E. LAUER,
IBM Laboratory Vienna,
Parkring 10,
A 1010 WIEN,
Austria.

The reference number of the report is TR 25.088 - Ed.).

AB31.3.2 Concerning the Syntax of ALGOL 68

C. Pair

1. General

1.1 Introduction

The "strict language" of ALGOL 68 is generated by a context-free grammar, generalised in the sense that its non-terminal vocabulary and its set of production rules are infinite. The non-terminal vocabulary is made up of "notions", which are particular cases of "protonotions".

The report on ALGOL 68* defines a protonotion as a non-empty, possibly infinite, sequence of "small syntactic marks". Undoubtedly, certain mode declarations give rise to notions which cannot be finite sequences of such signs. For example, this is so for the declaration:

(1) mode a = struct (ref a x)

But notions cannot be infinite sequences either. For if it were so, what would be obtained by replacing (R, 1.1.5), in "MODE source", the "metanotion" MODE by an infinite sequence? Where would "source" appear? Moreover, if one admits that the mode defined by the declaration (1) is an infinite sequence, this could not, with the given definitions, be other than:

structured with reference to structured with reference to structured .....

which will never contain x. Again, will this be the same mode as that defined by:

mode b = struct (ref b y, int z)?

No, of course not. On the other hand, it is tempting to think that (1) defines the same mode as:

mode c = struct (ref c x)

or even as the set of two declarations:

mode d = struct (ref e x); mode e = struct (ref d x)

Thus protonotions are not sequences.

1.2 Scheme of the Present Paper

The purpose of this paper is to study the nature of the set of protonotions. Paragraph 2 indicates an axiomatic definition of this set, codifying the properties which it is desired it will possess. Paragraph 3 defines the production rules of the strict language. Paragraph 4 justifies, in terms of the axioms, an algorithm which, in particular, allows the testing of two modes for equality; it can then be shown that the modes a, c, d considered above are equal and different from the mode b. Finally, paragraph 5 demonstrates the existence of a set satisfying the axioms proposed in paragraph 2, and its uniqueness up to isomorphism.

In particular, it is necessary to be able to form the product of two protonotions in order to give a sense to the phrase "in MODE source replace the metanotion MODE by one of its terminal productions": the protonotions will thus form a groupoid P, i.e. a set having an operation. This operation, or product, will be denoted by simple juxtaposition and,

---

* (See AB31.0) In what follows, this report will be referred to simply by R; and R, 1.1.5, for example, will refer to its section 1.1.5.

by convention, in the absence of parentheses, will be effected from left to right.

1.3 The Notion of Monomial

If L is a part of P and p a natural integer, let us consider the products of constants belonging to L and of p variables whose values are in P. For example, for p=3 and a, b in L

$$(a\ y_1\ y_2)\ (b\ y_3)\ a, \text{ that is } ((((a\ y_1)\ y_2)\ (b\ y_3))\ a),$$

is such a product.

More precisely, the following definition can be given. Let V be a set of p elements $y_1, \ldots, y_p$. The union of the sets V, L and of the set of two parentheses (,), supposed disjoint in pairs, constitutes the terminal vocabulary of the context-free language M defined by the equation:

$$M = L \cup V \cup (MM).$$

The words belonging to M are called monomials on L in p variables. If, in one of them, f say, $y_1, \ldots, y_p$ are replaced by p values $\alpha_1, \ldots, \alpha_p$ belonging to a groupoid P, a product of elements of P is obtained, whose value will be denoted by $f(\alpha_1, \ldots, \alpha_p)$: each monomial thus defines a function of p variables of P, whose values are in P, called a monomial function.

These functions extend to the set of parts of P. If $T_1, \ldots, T_p$ are subsets of P, $f(T_1, \ldots, T_p)$ denotes the set of $f(\alpha_1, \ldots, \alpha_p)$ where $\alpha_1 \in T_1, \ldots, \alpha_p \in T_p$.

2. Axiomatics of Protonotions

Let L be the set of small syntactic marks (R, 1.1.2,a). The set P of protonotions satisfies the axioms 1 and 2.

2.1 Axiom 1:

P is a groupoid containing L such that, if $\alpha_1, \alpha_2, \beta_1, \beta_2$ are elements of P:

(i) $\alpha_1 \alpha_2 = \beta_1 \beta_2$ implies $\alpha_1 = \beta_1$ and $\alpha_2 = \beta_2$;

(ii) $\alpha_1 \alpha_2 \notin L$.

Part (i), the necessity of which is less evident than that of part (ii), allows the assertion that if in "MODE field TAG selector", for example, we replace first MODE by μ and TAG by τ, then MODE by μ' and TAG by τ', we obtain identical results onlyfor $\mu = \mu'$ and $\tau = \tau'$.

It allows, moreover, the assertion that, whatever the protonotions which replace MODE, TAG, FIELD, FIELDS in "MODE field TAG" and "FIELDS and FIELD", equal products are never obtained. These two results are immediate consequences of the propositions 1 and 2 which we shall deduce from axiom 1.

Proposition 1: If $n \geqslant p$ and if $\alpha_1, \ldots, \alpha_n, \beta_1, \ldots \beta_p$ belong to P, the equality of the products $\alpha_1 \ldots \alpha_n$ and $\beta_1 \ldots \beta_p$ implies $\alpha_n = \beta_p, \ldots, \alpha_{n-p+2} = \beta_2, \alpha_1 \ldots \alpha_{n-p+1} = \beta_1$.

This is demonstrated immediately by recurrence upon p.

From this proposition and from part (ii) of axiom 1, we have the result that two products $a_1 \ldots a_n$ and $b_1 \ldots b_p$ of elements of L are not equal unless $n = p$ and $a_i = b_i$ for $1 \leqslant i \leqslant n$. Thus the set of non-empty words on the alphabet L can be identified with a part of P. But it must be noted that the concatenation of two words is distinct from their product in P, since (i) excludes the associativity of the product.

Proposition 2: Every monomial function is injective. By convention we consider as injective a function of zero variables. Let f be a monomial. We shall show that the function which it defines is injective, by recurrence on the length $|f|$ of the monomial f.

If $|f| = 1$, $f = a \in L$ defines a function of 0 variables, or $f = x_1$ defines the identity in P.

If $|f| > 1$, f is written in the form (f'f") and $f(\alpha_1, \ldots, \alpha_p) = f'(\alpha_{i_1}, \ldots, \alpha_{i_q})\ f''(\alpha_{j_1}, \ldots, \alpha_{j_r})$ where p, q, r are respectively the numbers of variables in f, f', f" and every integer from 1 to p figures among $i_1, \ldots, i_q$ or among $j_1, \ldots, j_r$. From axiom 1:

$$f(\alpha_1, \ldots, \alpha_p) = f(\beta_1, \ldots, \beta_p) \Longrightarrow f'(\alpha_{i_1}, \ldots, \alpha_{i_q})$$

$$= f'(\beta_{i_1}, \ldots, \beta_{i_q}) \text{ and}$$

$$f''(\alpha_{j_1}, \ldots, \alpha_{j_r}) = f''(\beta_{j_1}, \ldots, \beta_{j_r})$$

$$\Longrightarrow \alpha_{i_1} = \beta_{i_1} \text{ and } \ldots \text{ and } \alpha_{i_q} = \beta_{i_q} \text{ and}$$

$$\alpha_{j_1} = \beta_{j_1} \text{ and } \ldots \text{ and } \alpha_{j_r} = \beta_{j_r}$$

$$\Longrightarrow \alpha_1 = \beta_1 \text{ and } \ldots \text{ and } \alpha_p = \beta_p .$$

Since, by the hypothesis of recurrence, f' and f" define injective functions.

It can also be deduced from axiom 1, using this time part (ii) as well as part (i), that two distinct monomials define two distinct monomial functions.

2.2 System of Equations in P

Mode declarations such as those indicated in paragraph 1.1 lead to the definition of modes by means of equations or systems of equations whose second members are monomials. For example, if μ is the mode designated by a in the declaration (1) of paragraph 1.1, μ must satisfy the equation:

μ = structured with reference to μ field letter x.

We must admit that any mode - defining system possesses one solution and one solution only. Let us begin with systems whose equations are particularly simple.

Definition: A normal system is a system of n equations in n unknowns $x_1, \ldots, x_n$, where the ith equation $(1 \leq i \leq n)$ is of one of the forms

$$x_i = x_j\, x_k \quad \text{or} \quad x_i = a \quad \text{with} \quad a \in L.$$

Axiom 2: Every normal system admits one and only one solution with components in P, and every element of P belongs to the solution of a normal system.

Let us now generalise the notion of normal system and the result of axiom 2.

Definition: A proper system is a system of n equations in n unknowns, $x_1, \ldots, x_n$ where the ith equation $(1 \leq i \leq n)$ is of the form

$$x_i = f_i(x_{r_{i1}}, \ldots, x_{r_{ip}})$$

f_i being a monomial on L, and which is acyclic, that is to say, for every sequence of p (≥ 2) integers $j_1, \ldots, j_p$ such that $x_{j_1} = x_{j_2}, \ldots, x_{j_{p-1}}$ are equations of the system, j_p is different from j_1.

In particular, in an acyclic system, there is no equation $x_j = x_j$. A system which is not acyclic, and which admits one solution in P, necessarily admits an infinity of solutions, for the unknowns involved in a "cycle" have arbitrary values.

Proposition 3: Every proper system admits one and only one solution with components in P.

It is sufficient to transform the given system (S) into a normal system.

(a) The system (S) being acyclic, any equations $x_i = x_j$ can first be suppressed; indeed, the sequences $j_1, j_2 \ldots j_p$ of the preceding definition being of bounded length (bounded by the number of unknowns), for at least one of the equations $x_k = x_\ell$, the second member of the ℓth equation is not reduced to one unknown; a system (S') equivalent to (S) is obtained by replacing x by this second member in the equation $x_k = x_\ell$; the equations of (S') in which the second member is reduced to one unknown are those of (S), except for $x_k = x_\ell$; (S') is in turn acyclic and it is sufficient to repeat the same operation.

(b) In order to arrive at the forms of equation admitted in a normal system, any equation $x_i = pq$, where p and q are monomials not reduced to one unknown, is replaced by the three equations:

$$x_i = x'_i\, x''_i \qquad x'_i = p \qquad x''_i = q\ .$$

Similarly we replace

$$x_i = px_j \quad \text{by} \quad x_i = x'_i\, x_j,\ x'_i = p$$

and

$$x_i = x_j q \quad \text{by} \quad x_i = x_j\, x'_i,\ x'_i = q\ .$$

At each step the maximum length of the second members can be diminished so that the desired form is arrived at in a finite number of steps.

3. The Syntax of the Strict Language

3.1 The Metanotions

Each metanotion designates a subset of P, in a manner which will be specified.

(a) Modification of metaproduction rules (R, 1.2):

- replace NOTION: ALPHA; NOTION ALPHA.
  by NOTION: ALPHA; NOTION NOTION.

- transform the given rules into equivalent rules not involving the metanotion EMPTY.

The first transformation takes account of the fact that a "notion" is no longer a sequence of letters; it will be completely justified a little further down. The second is purely technical; it enables the presence of a neutral element in the groupoid P to be avoided - such an element, would complicate the axiomatic definition.

Each of the production rules of the metalanguage (R, 1.1.4) can then be written in the form

$$T_0 : f(T_1, \ldots, T_p)$$

where F is a monomial and T_0, T_1, ..., T_p are the metanotions which figure in the rule.

(b) Let us consider the proper systems

$$x_i = f_i(x_{r_{i1}}, \ldots, x_{r_{ip}}) \quad i = 1, \ldots, n$$

for which a sequence T_1, ..., T_n of n metanotions can be found such that, for all i, $T_i : f_i(T_{r_{i1}}, \ldots, T_{r_{ip}})$ is a production rule: we say then that the ith component of the solution is associated with T_i by the system. A metanotion <u>designates</u> the set of elements of P which are so associated with it by the preceding systems.

In particular, it is a result of the fact that every element of P belongs to the solution of a normal system (axiom 2), and of the rule

NOTION: ALPHA; NOTION NOTION.

that the metanotion NOTION designates the set P itself.

It is easily shown, besides, that the subsets designated by the metanotions satisfy the system (in the set of parts of P) comprising the equations obtained by replacing in the metaproduction rules ":" by "=", ";" by "$\cup$" and removing the final point.

3.2 The Production Rules of the Strict Language

They are obtained as prescribed in R, 1.1.5, if it is agreed:

(a) that a "terminal production" of a metanotion T is an element of the subset of P designated by T;

(b) to add to the end of 1.1.5:

step 4: in each of the sequences, formed of elements of L and of elements of P, replacing a metanotion, and which constitute the rule, place parentheses in some manner, to define a product in P.

(c) to add in 3.0.1, for example, the rule:

NOTION option: NOTION;.

4. Generation of a Program

4.1

In the generation of an actual program, use is made of a finite number of rules. The elements of P replacing the metanotions are finite

in number. They are given by a proper system (S), which can, as has been seen in the demonstration of porposition 3, be transformed into a normal system. The non-terminal symbols used are values of monomial functions, for (S); in other words, each non-terminal symbol is defined by adding an equation to (S). The essential problem of generation is to decide whether two non-terminal symbols so defined are equal, that is to say to test the equality of two components of the solution of a proper system.

In practice, moreover, equal components in the solution of (S) could be studied; this would enable the system to be simplified by removing any equations and unknowns which are used twice, so as to preserve only a system in which all the components of the solution are distinct.

The fundamental problem is thus: given a proper system, what are the components of its solution which are equal? We shall resolve it in the case where the system satisfies a supplementary hypothesis, always true for normal systems thanks to Axiom 1; the system can thus always be reduced to this case.

4.2 Study of the Problem of the Equality of Two Components of the Solution of a Proper System

$$\text{(S)} \qquad x_i = f_i\ (x_{r_{i1}}, \ldots, x_{r_{ip}}) \qquad i = 1, \ldots, n.$$

In the particular case of a normal system, each monomial f_i is either the monomial "product of two indeterminates" ($f_i(y_1,y_2) = y_1y_2$), or one of the monomials in zero indeterminates equal to an element of L. From axiom 1, a product cannot be equal to an element of L. Since, in the solution, the components x_i and x_j are equal, the monomials f_i and f_j are equal, the monomials f_i and f_j are <u>necessarily</u> equal.

In the general case we make this hypothesis:

<u>if in the solution of (S), the values of the unknowns x_i and x_j are equal, then the monomials f_i and f_j are equal.</u>

This hypothesis excludes, in particular, every equation of the form $x_i = x_j$, for, since the system is proper and so acyclic, there would necessarily exist one of these equations for which the monomial f_j which defines x_j would not reduce to an indeterminate. Besides, it can be shown that the modes entering into a program can be defined by a system which satisfies this hypothesis.

Thus in order that $x_i = x_j$ in the solution, it is necessary that $f_i = f_j$, and as the function f_i is injective (proposition 2), that

$$x_{r_{i1}} = x_{r_{j1}}, \ldots, x_{r_{ip}} = x_{r_{jp}}$$

(and these conditions are evidently sufficient).

Consider the graph whose vertices are the pairs $\{i,j\}$ of integers between 1 and n, and whose arcs with extremity $\{i,j\}$ have for origin, if $f_i = f_j$, the pairs $\{r_{i1},r_{j1}\}$, ..., $\{r_{ip},r_{jp}\}$ of indices of unknowns of the same rank in the ith and j^{th} equation; if $f_i \neq f_j$ no arc has $\{i,j\}$ as extremity: let E be the set of these pairs $\{i,j\}$ such that $f_i \neq f_i$ (whence necessarily $x_i \neq x_j$). In other words, if Γ is the relation of the graph, when $\{k,\ell\}\ \Gamma\{i,j\}$, $x_i = x_j$ implies $x_k = x_\ell$, $x_k \neq x_\ell$ implies $x_i \neq x_j$. Finally, let Γ * be the reflexive and transitive closure of the relation Γ.

<u>Proposition 4</u>. For the unknowns x_i and x_j, in the solution of the system, to have the same value, it is necessary and sufficient to have $\{i,j\} \notin \Gamma\ *\ (E)$.(†)

According to the previous definitions, $\{i,j\} \in \Gamma\ *(E)$ implies $x_i \neq x_j$. To study the converse, we shall first demonstrate a lemma involving only the definitions of E and of Γ, to the exclusion of all properties of the set P.

<u>Lemma</u>: The relation between the integers i and j defined by

$$\{i,j\} \notin \Gamma\ *(E)$$

is an equivalence.

This relation is evidently reflexive and symmetric. To justify its transitivity, let us demonstrate by recurrence on the integer q:

$$\{i,k\} \notin \Gamma\ *(E) \text{ and } \{k,j\} \notin \Gamma\ *(E) \Longrightarrow \{i,j\} \notin \Gamma^q(E).$$

The hypothesis of this proposition implies $\{i,k\} \notin E$ and $\{k,j\} \notin E$, whence $f_i = f_k = f_j$, whence $\{i,j\} \notin E$, which demonstrates the proposition for q = 0. The proposition being supposed true for q-1, let us effect a <u>reductio ad absurdum</u> by supposing that $\{i,j\}$ belongs to $\Gamma^q(E)$: there exist i' and j' such that:

$$\{i',j'\} \in \Gamma^{q-1}(E) \quad \text{and} \quad \{i',j'\}\ \Gamma\{i,j\};$$

i' and j' are the indices of unknowns of the same rank in the ith and j^{th} equations; let k' be the index of the unknown of the same rank in the kth equation:

$$\{i',k'\}\ \Gamma\{i,k\} \quad \text{and} \quad \{k',j'\}\ \Gamma\{k,j\}.$$

(†) In other words, there does not exist $\{k,\ell\} \in E$ such that $\{k,\ell\}\ \Gamma^*\{i,j\}$.

Thus $\{i,k\} \notin \Gamma^*(E)$ implies $\{i',k'\} \notin \Gamma^*(E)$

$\{k,j\} \notin \Gamma^*(E)$ implies $\{k',j'\} \notin \Gamma^*(E)$

Whence, by the recurrence hypothesis $\{i',j'\} \notin \Gamma^{q-1}(E)$, which is the contradiction sought for.

Let us return to the demonstration of proposition 4. We shall write $i \sim j$ when $\{i,j\} \notin \Gamma^*(E)$;

$i \sim j$ implies $f_i = f_j$ and $r_{i1} \sim r_{j1}, \ldots, r_{ip} \sim r_{jp}$.

Let us choose a representative in each equivalence class and construct a system (S') with:

- the equations obtained by replacing, in the equations of (S), each index by the representative of its class; a single equation is obtained for each class, for if $i \sim j$ the ith and j^{th} equations of (S) give rise to a single equation of (S');

- for each integer j whose class is represented by $i \neq j$, the equation $x_j = x_i$.

(S') is a proper system of n equations since, when it contains one equation $x_j = x_i$, the second member of the equation which defines x_i is not reduced to one unknown. Thus (S') possess one solution where $i \sim j$ implies $x_i = x_j$.

This solution is the unique solution of (S) : indeed let us consider the equation $x_j = f_j\ (x_{r_{j1}}, \ldots, x_{r_{jp}})$; if the representative of the class of j is i, $r_{j1} \sim r_{i1}, \ldots, r_{jp} \sim r_{ip}$ and (S') contain the equations:

$$\left.\begin{array}{l} x_i = f_i\ (x_{r'_{i1}}, \ldots, x_{r'_{ip}} \\ \\ x_j = x_i \\ \\ x_{r_{j1}} = x_{r'_{i1}} \\ \\ \ldots\ldots\ldots\ldots \\ \\ x_{r_{jp}} = x_{r'_{ip}} \end{array}\right\} \quad \text{Which imply } x_j = f_i\ (x_{r_{j1}}, \ldots, x_{r_{jp}})$$

It may be remarked that the test for equality of the components of the solution of a system, given by proposition 4, lends itself well to being programmed.

5. Existence and Uniqueness of a Groupoid P Satisfying Axioms 1 and 2

Proposition 5: There exists a groupoid P, unique up to isomorphism, which satisfies axioms 1 and 2.

We confine ourselves to an outline of the demonstration.

(1) Let us consider the set $\mathcal{S}$ of normal systems: it can be defined as a language on the alphabet formed of L, of an enumerable set $\{x_1, x_2, \ldots\}$ of unknowns and of the two signs = ; (; separating the equations). Given any two systems (S) and (S'), the juxtaposition of (S) and (S') consists of placing in sequence after (S) the system obtained by adding the number of unknowns of (S) to the indices of the unknowns of (S'): the juxtaposition is associative.

For a normal system, the equivalence of indices of two unknowns is defined as in the demonstration of proposition 4: the lemma of this proposition has indeed been justified without involving the existence of P. Moreover, it is shown that;

(a) if (S") is obtained by juxtaposition of (S) and (S'), and if i and j are indices of unknowns of (S), $i \sim j$ for (S") if and only if $i \sim j$ for (S);

(b) if (S') is derived from (S) by a permutation ϖ of unknowns and equations, $i \sim j$ for (S) if and only if $\varpi(i) \sim \varpi(j)$ for (S');

(c) if (S) is a normal system in n unknowns and if (S') is obtained by juxtaposition from (S) and (S), for all $i \leq n$, $i \sim i+n$.

(d) if $k \sim k'$ for (S) and if (S') is derived from (S) by replacing x_k by $x_{k'}$ in the second member of some equation, $i \sim j$ for (S') if and only if $i \sim j$ for (S).

(2) Let us suppose that P exists. Let φ be the application of $\mathcal{S}$ in P which associates with every system the first component of its solution: φ is a surjection according to axiom 2. φ will be a homomorphism of groupoids if the product of two normal systems (S), in n unknowns, and (S'), is defined as obtained by the juxtaposition of the equation $x_1 = x_2 x_{n+2}$, of (S), and of (S'). P is then isomorphic with the quotient of $\mathcal{S}$ by the equivalence relation associated with the homomorphism φ. Therefore if P exists, it is unique up to isomorphism.

(3) Conversely, let us define in $\mathcal{S}$ the product as has just been described, and an equivalence relation: a system (S), in n unknowns, is equivalent to a system (S') if, for the system obtained by juxtaposition of (S) and (S'), the indices 1 (of the first unknown of (S) and n+1 (of the first unknown of (S')) are equivalent. It is easily verified, using the properties (a), (b), (c), (d) enumerated above:

- that it has indeed the properties of an equivalence (reflexivity according to c, symmetry according to b, transitivity according to a and b);

- that it is compatible with the product (according to a);

- that the groupoid quotient of $\mathcal{S}$ by this equivalence relation satisfies axiom 1, if every element a of L is identified with the class of the system with a single equation $x_1 = a$ (according to the property a);

- that it also satisfies axiom 2: a solution of a system (S) is formed of classes of systems (S_1), ..., (S_n), where (S_i) is obtained by permuting in (S) the first and the ith unknown, the first and the ith equation (according to a, b, c); by writing besides that the classes of other system (S'_1) ..., (S'_n) also satisfy (S), a system is obtained which is transformed conformably with (d) in order to show that $(S'_1) \sim (S_i)$ for $i = 1, \ldots, n$.

Université de Nancy,
Centre de Calcul Automatique,
42 Avenue de la Libération,
Nancy.

ACKNOWLEDGEMENT: I am indebted to F.G. Duncan for his fair translation of my original French text. (received March 1969).